

EUROPEAN PATENT OFFICE

Patent Abstracts of Japan

PUBLICATION NUMBER : 2000122952
PUBLICATION DATE : 28-04-00

APPLICATION DATE : 16-10-98
APPLICATION NUMBER : 10295788

APPLICANT : FUJI XEROX CO LTD;

INVENTOR : IWATA NOBUO;

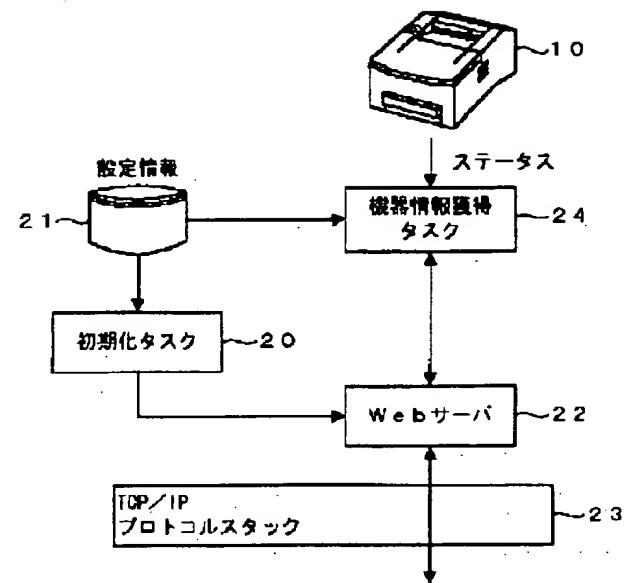
INT.CL. : G06F 13/00 G06F 3/00 G06F 3/12

TITLE : INFORMATION PROCESSOR

ABSTRACT : PROBLEM TO BE SOLVED: To display various pieces of information showing the states of various peripheral equipments with document display software and to easily process data.

SOLUTION: A Web server 22 acquires information to be outputted through a unit information acquirement task 24 and generates an HTTP(hyper text transfer protocol) telegram from information to be outputted in an HTML system or an XML (extensile mark up language) system or in two systems in accordance with an output system which is set in setting information 21. The HTTP telegram described in the generated HTML(hyper text mark up language) system or the XML system or in the two systems is transmitted. In a computer device, information of the HTML system is displayed with general reference software. In a server device, information of the XML system is analyzed with prescribed software and the state of the device and statistic information are data-processed and managed.

COPYRIGHT: (C)2000,JPO



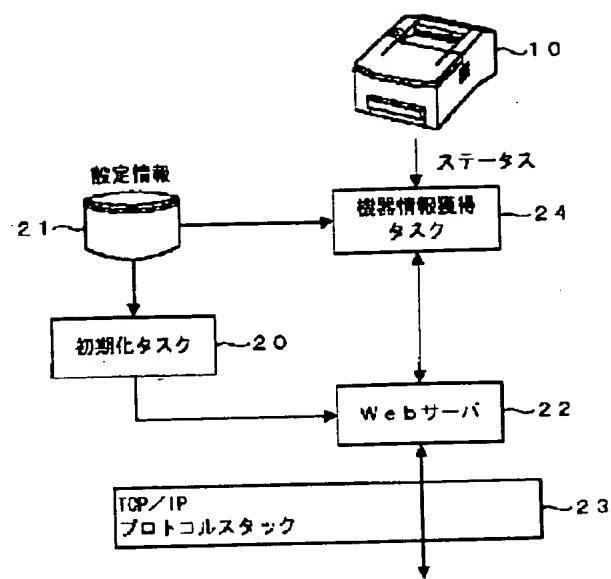
EUROPEAN PATENT OFFICE

Patent Abstracts of Japan

PUBLICATION NUMBER : 2000122952
PUBLICATION DATE : 28-04-00

APPLICATION DATE : 16-10-98
APPLICATION NUMBER : 10295788

APPLICANT : FUJI XEROX CO LTD;
INVENTOR : IWATA NOBUO;
INT.CL. : G06F 13/00 G06F 3/00 G06F 3/12
TITLE : INFORMATION PROCESSOR



ABSTRACT : PROBLEM TO BE SOLVED: To display various pieces of information showing the states of various peripheral equipments with document display software and to easily process data.

SOLUTION: A Web server 22 acquires information to be outputted through a unit information acquirement task 24 and generates an HTTP(hyper text transfer protocol) telegram from information to be outputted in an HTML system or an XML (extensile mark up language) system or in two systems in accordance with an output system which is set up language) 21. The HTTP telegram described in the generated HTML(hyper text in setting information 21. The HTTP telegram described in the generated HTML(hyper text mark up language) system or the XML system or in the two systems is transmitted. In a computer device, information of the HTML system is displayed with general reference software. In a server device, information of the XML system is analyzed with prescribed software and the state of the device and statistic information are data-processed and managed.

COPYRIGHT: (C)2000,JPO

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2001-22952

(P2001-22952A)

(43)公開日 平成13年1月26日(2001.1.26)

(51)Int.Cl'	識別記号	F I	マーク(参考)
G 0 6 T 15/40	2 0 0	G 0 6 T 15/40	2 0 0
G 0 6 F 3/14	3 1 0	G 0 6 F 3/14	3 1 0 A
G 0 9 G 5/36	5 1 0	G 0 9 G 5/36	5 1 0 V

審査請求 有 請求項の数32 OL (全21頁)

(21)出願番号	特願2000-148191(P2000-148191)	(71)出願人	390009531 インターナショナル・ビジネス・マシーンズ・コーポレーション INTERNATIONAL BUSINESS MACHINES CORPORATION アメリカ合衆国10504、ニューヨーク州 アーモンク(番地なし)
(22)出願日	平成12年5月19日(2000.5.19)	(74)代理人	100086243 弁理士 坂口 博(外2名)
(31)優先権主張番号	09/321787		
(32)優先日	平成11年5月27日(1999.5.27)		
(33)優先権主張国	米国(US)		

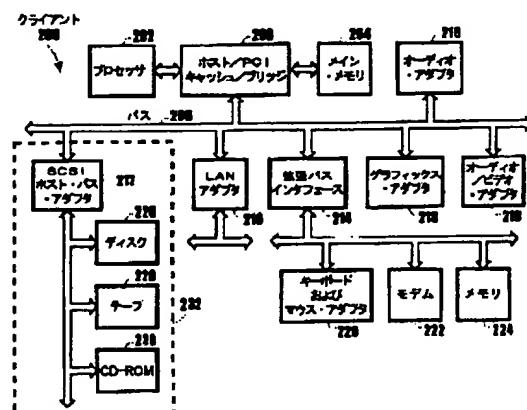
最終頁に続く

(54)【発明の名称】 描画を表示する方法及びデータ処理システム

(57)【要約】

【課題】 遮断されるオブジェクトを識別する方法及び装置を提供する。

【解決手段】 視点のために描画を表示するデータ処理システムにおける方法及び装置。ここで、描画は一組のオブジェクトを含む。オブジェクトの組のための複数の境界ボックス及び複数データが受け取られる。ここで、境界ボックス及び複数データは、オブジェクトの組の中の各オブジェクトに関連づけられている。複数の境界ボックスを使用して、オブジェクトの組の中の遮断オブジェクトが選択される。遮断オブジェクトと比較されたとき、視点から可視であるオブジェクトが、オブジェクトの組から識別される。



【特許請求の範囲】

【請求項1】 視点のために描画を表示するデータ処理システムにおける方法であって、ここで描画は一組のオブジェクトを含み、該方法は、オブジェクトの組のために複数の境界ボックス及び複雑度データを受け取り、ここで前記境界ボックス及び前記複雑度データはオブジェクトの組内の各オブジェクトに関連づけられ、複数の前記境界ボックスを使用して前記オブジェクトの組内の遮断オブジェクトを選択し、

視点のために可視のオブジェクトを前記オブジェクトの組から識別するコンピュータ実現ステップを含む方法。

【請求項2】 前記オブジェクトが識別されたオブジェクトであり、前記識別されたオブジェクトを表示するステップを更に含む、請求項1に記載の方法。

【請求項3】 ピクセル深度データが各前記遮断オブジェクトに関連づけられ、且つ可視であるオブジェクトの組からオブジェクトを識別するステップが、前記オブジェクトの組内のオブジェクトの前記境界ボックスの深度が、前記遮断オブジェクトのピクセルの深度よりも視点に近いかどうかを決定することを含む、請求項1に記載の方法。

【請求項4】 前記ピクセル深度データが、前記ピクセル深度データを発生するための低解像度で深度情報を用いて前記遮断オブジェクトを十分詳細にレンダリングすることによって発生する、請求項3に記載の方法。

【請求項5】 前記遮断オブジェクトを選択するステップが、前記オブジェクトの組を複雑度によってソートし、前記オブジェクトの組から、描画の複雑度の設定パーセンテージよりも小さなトータル複雑度を有する或る数のオブジェクトを選択することを含む、請求項1に記載の方法。

【請求項6】 前記オブジェクトの数が255よりも少ない、請求項5に記載の方法。

【請求項7】 前記設定パーセンテージが3.5パーセントである、請求項6に記載の方法。

【請求項8】 前記推定するステップが、前記オブジェクトの組の中で、全体的な最大可視領域、最も近い深度値、及び最低複雑度値を有するオブジェクトを識別することを含む、請求項1に記載の方法。

【請求項9】 前記複雑度データが、オブジェクトをレンダリングするのに必要な時間を含む、請求項1に記載の方法。

【請求項10】 前記遮断オブジェクトのトータル複雑度が残りの部分の複雑度よりも大きいかどうかを決定することを更に含む、請求項1に記載の方法。

【請求項11】 前記他のオブジェクトが可視であるかどうかを決定するステップが、前記遮断オブジェクトの複

雑度が残りの部分の複雑度よりも小さいことの決定に応答する、請求項10に記載の方法。

【請求項12】 選択、識別、及び表示ステップを実行するために使用される第1の時間量を決定し、選択及び識別ステップを実行することなくオブジェクトの組を表示するために使用される第2の時間量を決定し、

第1の時間量が第2の時間量よりも大きいことに応答して、オブジェクトの組を表示するとき選択及び識別ステップの実行を防止することを更に含む、請求項2に記載の方法。

【請求項13】 視点のためにシーンを処理するデータ処理システムにおける方法であって、ここでシーンは複数のオブジェクトを含み、該方法は、シーンのためのデータを受け取り、シーンのためのデータを使用して前記複数のオブジェクトから一組の可視オブジェクトを選択し、前記複数のオブジェクトから選択されないオブジェクトを、前記可視オブジェクトの組と比較することによって、視点のために前記可視のオブジェクトを識別するデータ処理システムのステップを含む方法。

【請求項14】 データが、前記複数のオブジェクトの境界ボックス、及び前記複数のオブジェクトの複雑度データを含む、請求項13に記載の方法。

【請求項15】 ピクセル深度データが各前記可視オブジェクトに関連づけられ、オブジェクトの組からの他のオブジェクトが可視であるかどうかを識別するステップが、

前記境界ボックスの深度が前記遮断オブジェクトのピクセルの深度よりも視点に近いかどうかを決定することを含む、請求項14に記載の方法。

【請求項16】 視点のために描画を表示するデータ処理システムであって、ここで描画は一組のオブジェクトを含み、該データ処理システムは、

オブジェクトの組のために複数の境界ボックス及び複雑度データを受け取る受け取り手段と、ここで前記境界ボックス及び前記複雑度データはオブジェクトの組内の各オブジェクトに関連づけられ、複数の前記境界ボックスを使用して、前記オブジェクトの組内の遮断オブジェクトを選択する選択手段と、前記オブジェクトの組から、視点のために可視のオブジェクトを識別する識別手段とを含むデータ処理システム。

【請求項17】 前記オブジェクトが識別されたオブジェクトであり、前記識別されたオブジェクトを表示する表示手段を更に含む、請求項16に記載のデータ処理システム。

【請求項18】 ピクセル深度データが各前記遮断オブジェクトに関連づけられ、且つ前記可視のオブジェクトをオブジェクトの組から識別するステップが、

前記オブジェクトの組内のオブジェクトの境界ボックスの深度が、前記遮断オブジェクトのピクセルの深度よりも視点に近いかどうかを決定する決定手段を含む、請求項16に記載のデータ処理システム。

【請求項19】前記ピクセル深度データが、ピクセル深度データを発生するための低解像度で深度情報を用いて前記遮断オブジェクトを十分詳細にレンダリングすることによって発生する、請求項18に記載のデータ処理システム。

【請求項20】前記遮断オブジェクトを選択する選択手段が、

前記オブジェクトの組を複雑度によってソートするソート手段と、

前記オブジェクトの組から、描画の複雑度の設定パーセンテージよりも小さいトータル複雑度を有する或る数のオブジェクトを選択する選択手段とを含む、請求項16に記載のデータ処理システム。

【請求項21】前記オブジェクトの数が255より少ない、請求項20に記載のデータ処理システム。

【請求項22】前記設定パーセンテージが3.5パーセントである、請求項21に記載のデータ処理システム。

【請求項23】前記推定手段が、

前記オブジェクトの組の中で、全体的な最大可視領域、最も近い深度値、及び最低複雑度値を有するオブジェクトを識別する識別手段を含む、請求項16に記載のデータ処理システム。

【請求項24】前記複雑度データが、オブジェクトをレンダリングするのに必要な時間を含む、請求項16に記載のデータ処理システム。

【請求項25】前記遮断オブジェクトのトータル複雑度が残りの部分の複雑度よりも大きいかどうかを決定する決定手段を更に含む、請求項16に記載のデータ処理システム。

【請求項26】前記他のオブジェクトが可視であるかどうかの決定手段が、前記遮断オブジェクトの複雑度が残りの部分の複雑度よりも小さいことの決定に応答する、請求項25に記載のデータ処理システム。

【請求項27】選択、識別、及び表示ステップを実行するために使用される第1の時間量を決定する第1の決定手段と、

選択、及び識別ステップを実行することなく前記オブジェクトの組を表示するために使用される第2の時間量を決定する第2の決定手段と、

前記第1の時間量が前記第2の時間量よりも大きいことに応答して、前記オブジェクトの組を表示するとき選択及び識別手段の制限を防止する防止手段とを更に含む、請求項17に記載のデータ処理システム。

【請求項28】視点のためにシーンを処理するデータ処理システムであって、ここでシーンは複数のオブジェクトを含み、該データ処理システムは、

シーンのためのデータを受け取る受け取り手段と、シーンのためのデータを使用して、複数のオブジェクトから一組の可視オブジェクトを選択する選択手段と、前記複数のオブジェクトから選択されないオブジェクトを、前記可視オブジェクトの組と比較することによって、視点のために前記可視のオブジェクトを識別する識別手段とを含む。

【請求項29】前記データが、複数のオブジェクトのための境界ボックス、及び複数のオブジェクトのための複雑度データを含む、請求項28に記載のデータ処理システム。

【請求項30】ピクセル深度データが各可視オブジェクトに関連づけられ、前記オブジェクトの組からの他のオブジェクトが可視であるかどうかを識別するステップが、

前記境界ボックスの深度が、遮断オブジェクトのピクセルの深度よりも視点に近いかどうかを決定する決定手段を含む、請求項29に記載のデータ処理システム。

【請求項31】視点のために描画を表示するための、コンピュータ読み取り可能媒体におけるコンピュータ・プログラム製品であって、ここで描画は一組のオブジェクトを含み、該コンピュータ・プログラム製品は、オブジェクトの組のために複数の境界ボックス及び複雑度データを受け取る第1の命令と、ここで前記境界ボックス及び前記複雑度データは、オブジェクトの組の中の各オブジェクトに関連づけられ、

複数の前記境界ボックスを使用して前記オブジェクトの組の中の遮断オブジェクトを選択する第2の命令と、前記オブジェクトの組から、視点のために可視のオブジェクトを識別する第3の命令とを含むコンピュータ・プログラム製品。

【請求項32】視点のためにシーンを処理するための、コンピュータ読み取り可能媒体におけるコンピュータ・プログラム製品であって、ここで前記シーンは複数のオブジェクトを含み、前記コンピュータ・プログラム製品は、

シーンのためのデータを受け取る第1の命令と、シーンのためのデータを使用して、複数のオブジェクトから一組の可視オブジェクトを選択する第2の命令と、前記複数のオブジェクトから選択されないオブジェクトを、前記可視オブジェクトの組と比較することによって、視点のために可視のオブジェクトを識別する第3の命令とを含むコンピュータ・プログラム製品。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、一般的には改善されたデータ処理システムに関し、具体的にはデータ処理システムで描画をレンダリングする改善された方法及び装置に関する。更に具体的には、本発明は、データ処理システムにおいて、目に見えるオブジェクト及び遮断さ

れるオブジェクトを識別する方法及び装置に関する。

【0002】

【従来の技術】一般的に、3Dコンピュータ・グラフィックスの目的は、ディスプレイ・スクリーン上で、3次元モデル空間内の所定の視点から展望したような3次元モデルの2D投影を作成することである。そのような投影の1つの局面は、視点から展望したとき、どのオブジェクトが他のオブジェクトの前方にあるか、及び、どれが後方にあるかを追跡する必要性がある。この知識は、例えば、前景にあるビルが遠方にあるビルを適正に遮断することを保証するために必要である。レンダリング・プロセスのこの局面は、「遮断選別」として知られている。

【0003】遮断選別を実行する1つの一般的な手法は、「zバッファ」として知られるコンストラクトを使用する。標準のzバッファは、観察者からの距離(投影面に対するシーンの深度)を表す「z値」と呼ばれる数を、スクリーン上に描画される各ピクセルに関連づける。シーンの最初のオブジェクトが投影されるとき、そのピクセルの属性(色など)が「フレーム・バッファ」にストアされ、各ピクセルに関連づけられたz値は、zバッファに別個にストアされる。もしその後で、モデルからの2番目のオブジェクトが、最初のオブジェクトからのデータを含むピクセルの上に投影されると、2番目のオブジェクトのz値が、そのピクセルについて既にストアされているz値と比較される。2番目のオブジェクトのz値が小さいときにのみ(オブジェクトが視点へより近いことを表す)、新しいピクセル及びzバッファが、2番目のオブジェクトの属性で更新される。

【0004】モデル空間でゼロに等しいzの所に置かれた投影平面又はイメージ平面で、異なった正の深度に置かれた2つのオブジェクト、オブジェクト1及びオブジェクト2をレンダリングする場合を考える。オブジェクト1が最初に投影されてレンダリングされる。2番目に、オブジェクト2がレンダリングされる。zバッファは、オブジェクト1がより小さいz値でピクセルを既に書き込んだフレーム・バッファの位置へ、オブジェクト2のピクセルが書き込まれないように防止する。従って、オブジェクト2は、所望のように、オブジェクト1の背後で最後に表示されるイメージとして現れる。

【0005】zバッファは、ハードウェア又はソフトウェアのいずれによっても実現することができる。ストアされる数は、浮動小数点数又は整数のいずれであってもよい。z値には、任意数のビットを使用することができる。一般的に、z値をストアするために使用されるビットが多ければ、それだけ精密な距離の解像度が達成される。z値はシーンにおけるオブジェクトの深度を表すから、z値は、より一般的には「深度値」と呼ぶことができる。zバッファは、より一般的には「深度バッファ」と呼ぶことができる。更に、深度が増大するとき、深度値

を増大することができ、又は、深度が減少するとき、深度値を減少することができる。遮断されるオブジェクト及び遮断されないオブジェクトを識別するために使用されるプロセスの多くは、時間を取り、シーンをレンダリングするのに必要な時間を増大する。

【0006】

【発明が解決しようとする課題】従って、遮断されるオブジェクトを識別する改善された方法及び装置を有することには、利点がある。

【0007】

【課題を解決するための手段】本発明は、描画が一組のオブジェクトを含むデータ処理システムにおいて、視点のために描画を表示する方法を提供する。オブジェクトの組のために、複数の境界ボックス及び複雑度データが受け取られる。ここで、境界ボックス及び複雑度データは、オブジェクトの組内の各オブジェクトに関連づけられている。オブジェクトの組内の遮断オブジェクトは、複数の境界ボックス及び複雑度データを使用して選択される。これらの遮断オブジェクトは、視点のためにオブジェクトの組から可視オブジェクトを識別するために使用される。

【0008】

【発明の実施の形態】ここで、図面、特に図1を参照すると、本発明が実現されてよいデータ処理システムの図が、本発明の好ましい実施形態に従って描かれている。パーソナル・コンピュータ100はシステム・ユニット110、ビデオ表示端末102、キーボード104、記憶装置108、及びマウス106を含むように描かれる。記憶装置108はフロッピー(登録商標)・ドライブ及び他のタイプのパーマネント及び取り外し可能記憶媒体を含んでよい。追加の入力装置が、パーソナル・コンピュータ100に含まれてよい。パーソナル・コンピュータ100は、IBM社(International Business Machines Corporation, located in Armonk, New York)の製品であるIBM Aptiva(商標)コンピュータのような任意適切なコンピュータを使用して実現されることができる。描かれた表現はパーソナル・コンピュータを示しているが、本発明の他の実施形態は、ネットワーク・コンピュータ、ウェブ・ベースのテレビジョン・セット・トップ・ボックス、インターネット機器など、他のタイプのデータ処理システムで実現されてよい。更に、コンピュータ100は、好ましくは、コンピュータ100の中で動作するコンピュータ読み取り可能媒体に存在するシステム・ソフトウェアによって実現されてよいグラフィカル・ユーザ・インターフェースを含む。

【0009】ここで図2を参照すると、ブロック図は、本発明が実現されてよいデータ処理システムを示す。データ処理システム200は、図1のコンピュータ100のように、本発明のプロセスを実現するコード又は命令が置かれてよいコンピュータの例である。データ処理シ

システム200はPCI (peripheral component interconnect) ローカル・バス・アーキテクチャを使用する。描かれた例はPCIバスを使用するが、マイクロチャネル及びISAのような他のバス・アーキテクチャを使用してよい。プロセッサ202及びメイン・メモリ204は、PCIブリッジ208を介してPCIローカル・バス206へ接続される。更に、PCIブリッジ208は、プロセッサ202のために統合化されたメモリ・コントローラ及びキャッシング・メモリを含んでよい。PCIローカル・バス206への追加の接続は、直接コンポーネント相互接続又はアドイン・ボードを介してなされてよい。描かれた例では、ローカル・エリア・ネットワーク (LAN) アダプタ210、SCSIホスト・バス・アダプタ212、及び拡張バス・インターフェース214が、直接コンポーネント接続によってPCIローカル・バス206へ接続される。対照的に、オーディオ・アダプタ216、グラフィックス・アダプタ218、及びオーディオ/ビデオ・アダプタ219は、拡張スロットへ挿入されたアドイン・ボードによってPCIローカル・バス206へ接続される。拡張バス・インターフェース214は、キーボード及びマウス・アダプタ220、モデム222、及び追加メモリ224のために接続を提供する。SCSIホスト・バス・アダプタ212は、ハード・ディスク・ドライブ226、テープ・ドライブ228、及びCD-ROMドライブ230のために接続を提供する。通常のPCIローカル・バス実現方式は、3つ又は4つのPCI拡張スロット又はアドイン・コネクタをサポートするであろう。

【0010】オペレーティング・システムはプロセッサ202上で実行され、図2のデータ処理システム200内で各種のコンポーネントを調整し且つ制御を提供するために使用される。オペレーティング・システムは、IBM社から入手できるOS/2のように、市販されているオペレーティング・システムであってよい。「OS/2」はIBM社の商標である。Javaのようなオブジェクト指向プログラミング・システムがオペレーティング・システムと組み合わせて実行されてよく、データ処理システム200上で実行されているJavaプログラム又はアプリケーションからオペレーティング・システムへの呼び出しを提供する。「Java」はサン・マイクロシステムズ社 (Sun Microsystems, Inc.) の商標である。オペレーティング・システム、オブジェクト指向プログラミング・システム、及びアプリケーション又はプログラムの命令は、ハード・ディスク・ドライブ226のような記憶装置上に置かれ、プロセッサ202によって実行されるためメイン・メモリ204へロードされてよい。

【0011】当業者は、図2のハードウェアが実現方式に従って変化してよいことを理解するであろう。フラッシュROM (又は同等の不揮発性メモリ) 又は光学ディスク・ドライブなどの他の内部ハードウェア又は周辺装

置が、図2に描かれたハードウェアに加えて、又はその代わりに、使用されてよい。更に、本発明のプロセスは、遮断選別プロセスを実行するためにマルチスレッド並列処理が使用されるマルチプロセッサ・データ処理システムへ応用されてよい。

【0012】例えば、データ処理システム200は、もしオプションのネットワーク・コンピュータとして構成されるならば、オプションとして含まれることを示す図2の点線232で示されるように、SCSIホスト・バス・アダプタ212、ハード・ディスク・ドライブ226、テープ・ドライブ228、及びCD-ROM230を含まなくともよい。その場合、コンピュータは、クライアント・コンピュータと呼ばれるのが適切であり、LANアダプタ210、モデム222などの或るタイプのネットワーク通信インターフェースを含む必要がある。他の例として、データ処理システム200は、或るタイプのネットワーク通信インターフェースを含むか含まないかに問わらず、或るタイプのネットワーク通信インターフェースに依存しないでポート可能に構成されるスタンダードアロン・システムであってよい。更に、他の例として、データ処理システム200は、オペレーティング・システム・ファイル、及び、又は、ユーザ生成データを記憶する不揮発性メモリを提供するために、ROM、及び、又は、フラッシュROMで構成されるパーソナル・ディジタル・アシスタント (PDA) 装置であってよい。

【0013】図2に描かれた例、及び前述した例は、アーキテクチャ上の制限を意味するものではない。

【0014】次に図3を参照すると、グラフィックス処理システムのシステム・アーキテクチャのブロック図が、本発明の好ましい実施形態に従って描かれる。グラフィックス処理システム300は処理ユニット302、ラスター・エンジン304、深度バッファ付きパック・フレーム・バッファ306、及び深度バッファ付きフロント・フレーム・バッファ308を含む。深度バッファ付きパック・フレーム・バッファ306及び深度バッファ付きフロント・フレーム・バッファ308は、この例ではダブル・バッファ・グラフィックス・システムを形成する。実現方式に依存して、深度バッファ付きパック・フレーム・バッファ306及び深度バッファ付きフロント・フレーム・バッファ308の代わりに、パチャル・フレーム・バッファを使用してよい。このタイプの実現方式では、遮断選別プロセスは、グラフィックス・アダプタが存在しないデータ処理システムで実行されてよい。結果は、グラフィックス・アダプタを有する他のデータ処理システムへ送られてよい。この例では、本発明の遮断選別プロセスは、描画又はシーンの実際の表示がクライアントによって実行される間に、サーバによって実行されてよい。

【0015】処理ユニット302は、処理のためのグラフィックス・データ・ストリーム又は表示リストを受け

取ってよい。処理ユニット302は1つ又は複数のプロセッサを含んでよい。処理ユニット302は、プリミティブが処理されてラスター・エンジン304へ送られるジョメトリ・エンジンの形式を取ってよい。ラスター・エンジン304は、処理されたプリミティブを表示用のピクセルへ変換する。深度バッファ付きパック・フレーム・バッファ306及び深度バッファ付きフロント・フレーム・バッファ308は、ピクセルのために赤、緑、及び青(RGB)を示すピクセル・データ・セットのようなデータをストアするために使用されるメモリから形成される。更に、これら2つのバッファは、オブジェクトの可視性を決定するために使用される深度データとも呼ばれるz軸データをストアする。これらのバッファ又はパック・バッファ(図示されていない)は、本発明の好ましい実施形態に従って、レンダリングされたオブジェクトを、小さなビューポートにストアするために使用される。処理ユニット302は、データ処理システム内の各種のプロセッサを使用して実現されてよい。例えば、図2のプロセッサ202が、処理ユニット302の機能を実行するために使用されてよい。代替的に、図2のグラフィックス・アダプタ218内のプロセッサが、処理ユニット302を実現するために使用されてよい。

【0016】深度バッファ付きパック・フレーム・バッファ306及び深度バッファ付きフロント・フレーム・バッファ308は、図2のグラフィックス・アダプタ218のようなグラフィックス・アダプタ内のメモリに置かれてよい。更に、これら2つのバッファは、図2のメイン・メモリ204のようなデータ処理システム内のメイン・メモリの中で実現されてよい。

【0017】ここで、図4～図7を参照すると、フ拉斯タム(frustum)内のオブジェクトのビューが、本発明の好ましい実施形態に従って描かれる。これらの図に示されるオブジェクトは、本発明の遮断選別プロセスが実現されてよいオブジェクトである。図4では、フ拉斯タム400の等距離ビューが、視点412に関して示される。頭を切り取られたピラミッドであるフ拉斯タム400は、オブジェクト402～410を含む。フ拉斯タム400は、近い平面414及び遠い平面416を含む。近い平面414は、視点412に最も近い平面である。観察者が正のz方向でz軸を見下ろす原点に位置しているアイ座標系(eye coordinate system)では、視野はフ拉斯タム400によって限定される。近い平面414は前方クリップ面とも呼ばれ、遠い平面416は後方クリップ面とも呼ばれる。

【0018】図5～図7には、フ拉斯タム400及びオブジェクト402～410の異なったビューが示される。これらの例では、オブジェクト402～406はオブジェクト408及び410の前方に現れる。オブジェクト408はオブジェクト410の前方にある。従つて、視点412からは、オブジェクト408及び410

は、図6から分かるように、隠されているか遮断されている。これらの例では、オブジェクト402～406は、オブジェクト408及び410よりも視点412に近い。結果として、オブジェクト402～404のz又は深度値は、オブジェクト408及び410のz又は深度値よりも小さい。

【0019】本発明は、図4～図7のシーンのようなシーンで、オブジェクトが可視であるかどうかを決定する方法、装置、及び命令を提供する。本発明のメカニズムによって、与えられたビューからオブジェクトの最小セットを決定することができる。本発明のメカニズムは、各オブジェクトに関連づけられた境界ボックス及び複雑度データを入力として受け取る。

【0020】「境界ボックス」とは、軸に整列した最小3次元平行六面体である。この六面体はオブジェクト、又は1つ又は複数のポリゴン、ポリゴン・エッジ、又は点を囲んでいる。境界ボックスの最小及び最大x、y、並びにz座標は、境界ボックスによって囲まれたオブジェクト、ポリゴン、ポリゴン・エッジ、又は点のそれぞれ最小及び最大x、y、並びにz座標に等しい。「投影された境界ボックス」とは、ビュー平面で、投影されたオブジェクト、投影されたポリゴン、投影されたポリゴン・エッジ、又は投影された点を囲む軸に整列した2次元長方形である。投影された境界ボックスの最小及び最大x及びy座標は、投影された境界ボックスによって囲まれる投影されたオブジェクト、投影されたポリゴン、投影されたポリゴン・エッジ、又は投影された点のそれぞれ最小及び最大の、x及びy座標に等しい。

【0021】境界ボックスは、遮断オブジェクト(可視オブジェクト)になる可能性があるシーン又は描画のオブジェクトについて、迅速で粗い推定を提供する。一度、遮断オブジェクトが識別されると、本発明のメカニズムは、描画又はシーンの現在のビューの中で、他のオブジェクトが可視であるかどうかを決定する。しかし、他のオブジェクトの処理へ進む前に、本発明を継続する価値があるかどうかについて、決定がなされる。描かれた例では、この決定は入力複雑度データに基づいている。

【0022】オブジェクトの複雑度値は、そのレンダリング時間に比例する。もし遮断オブジェクトのトータル複雑度が、残りのオブジェクトのトータル複雑度よりも大きければ、本発明のメカニズムは、オブジェクトの可視性を検査して追加の時間を費やすよりも、全アセンブリをレンダリングするのが早いと想定して終了する。この場合、全体の描画又はシーンに関してビューから隠されるオブジェクトの数は、確率的に小さい。逆に、もし遮断オブジェクトのトータル複雑度が、残りのオブジェクトのトータル複雑度よりもはるかに小さければ、多くのレンダリング時間を節約することのできる可能性がある。この状況に応答して、本発明のメカニズムは進行す

る。

【0023】オブジェクトの可視性は、先ず、全ての遮断オブジェクトを、深度情報で十分詳細に低解像度でレンダリングし、結果のピクセル深度データをストアすることによって検査される。新しい境界ボックスがオブジェクトのために作成され、その境界ボックスが現在のビューと軸的に整列するようにされる。次に、新しい境界ボックスの最も近い壁がシーンへ投影される。この投影は、投影された境界ボックスである。「近さ」は、視点に関する近さである。壁の深度は、前述したようにして得られた遮断オブジェクトの深度データと比較される。もし壁が、壁の投影された境界ボックス内のピクセルで、遮断オブジェクトの深度値よりも視点に近ければ、オブジェクトは可視である。

【0024】遮断オブジェクトは、全体的な最大の可視領域（境界ボックスに基づいて）、視点へ最も近い深度値、及び最低複雑度値を有する部品を選択することによって、知能的に選ばれてよい。

【0025】レンダリング時間は、入力部品データを静的と宣言することによって、同じシーンの将来のフレーム中に節約することができる。もし視点が2つ以上の連続フレーム中に著しく変化しないのであれば、それらフレームで遮断オブジェクトを使用してよい。

【0026】描かれた例では、グラフィックス・データ及びシステムは、3次元グラフィックスをレンダリングするライブラリであるOpenGLを使用して説明される。OpenGLはシリコン・グラフィックス社 (Silicon Graphics Incorporated) の商標である。オブジェクトは表示リストの形式で受け取られる。表示リストはOpenGLコマンドの指名リストである。表示リストの内容は事前に処理されていてよく、従って、即時モードで実行されるOpenGLコマンドの同じセットよりも効率的に実行されよう。

【0027】ここで、図8を参照すると、所与のビューから見ることのできるオブジェクトの最小セットを識別するプロセスのフローチャートが、本発明の好ましい実施形態に従って描かれる。このプロセスは、遮断オブジェクトの識別、及び所与のシーン又は描画について他の全てのオブジェクトの可視性を決定するためにそれら遮断オブジェクトを使用することに基づいて、遮断選別を提供する。

【0028】プロセスは、オブジェクト表示リストの配列、オブジェクト境界ボックスの配列、オブジェクト複雑度の配列、配列の長さ（入力オブジェクトの総数に等しい）、オブジェクト・リストが修正されたかどうかを示す新規シーン・フラグ、及びビューポートを入力として受け取る。このプロセスからの出力は、オブジェクト表示リストの入力配列を参照する可視オブジェクト・インデックスの配列、及び可視オブジェクトの数である。

【0029】プロセスは、新規シーン・フラグがTRUEで

あるかどうかをチェックすることによって開始する（ステップ500）。もし新規シーン・フラグがTRUEであれば、データ構造は、新しいシーンのために初期化される（ステップ502）。これらのデータ構造は、可視状況を各オブジェクトに関連づける。境界ボックス・データ及び複雑度データも、境界ボックス表面領域値と同じように、これらのデータ構造に含まれてよい。データ構造は、配列又はリンク・リストとして実現されてよい。次に、現在のグラフィックス状態が保存される（ステップ504）。グラフィックス状態が保存されるのは、この例では、遮断処理を実行するために、グラフィックス処理システムが使用されるからである。そうでなければ、この処理は、ユーザーによって観察されるグラフィカル表示を中断する。遮断処理が終了すると、グラフィックス状態がリストアされ、ユーザーはディスプレイ上で所望のシーン以外は見えないようにされる。もし新規シーン・フラグが「FALSE」であれば、プロセスは前述したステップ504へ直接進む。

【0030】次に、2つの静的フラグ、get_new_occluders及びcull_worthyが宣言され、get_new_occludersが「TRUE」値へ初期化される（ステップ506）。フラグget_new_occludersは、プロセスが実行される最初でのみ、「TRUE」値へ初期化される静的フラグである。次に、新規シーン・フラグがTRUEであるかどうか、又はget_new_occludersがTRUEであるかどうかの決定がなされる（ステップ508）。もし新規シーン・フラグがTRUEであるか、又はget_new_occludersがTRUEであれば、遮断オブジェクトが選択される（ステップ510）。cull_worthy静的フラグは、遮断オブジェクトの選択ステップ中に、「TRUE」又は「FALSE」値のいずれかへセットされる。このステップは、後の図9により詳細に説明される。

【0031】その後で、cull_worthyの値に基づいて、プロセスを継続する値があるかどうかの決定がなされる（ステップ512）。もしプロセスを継続する値があれば（cull_worthyは「TRUE」値を有する）、遮断オブジェクトは、パック・パッファの小さなビューポートにある深度バッファへのみ（色又はライティングはストアされない）レンダリングされる（ステップ515）。ビューポートとは、ピクセルが表示されるフレーム・パッファ領域である。これは「遮断検出ビューポート」である。プロセスのパフォーマンスを増大するために、小さなビューポートが選択される。なぜなら、アクセスしなければならないピクセルが少なくなるからである。プロセスのこのステップでは、深度情報のみが必要な情報であるから、ライティングの計算は抑止される。このレンダリングの結果として、遮断オブジェクトの深度値がストアされる。もし遮断オブジェクトがオーバラップすれば、眼に最も近い遮断オブジェクトの深度値がストアされる。

【0032】次に、ステップ510で遮断オブジェクトとして選択されなかった全てのオブジェクトについて、遮断のテストがなされる（ステップ516）。次に、フラグ`get_new_occluders`がトグルされる（ステップ518）。このフラグのトグルの結果として、シーンの1つおきの表示でのみ、遮断オブジェクトが選択される。もちろん、実現方法によっては、プロセスを毎回実行してよい。例えば、もしオブジェクトの滑らかな回転が起こるならば、描かれた例では、遮断プロセスは1つおきのフレームで実行されてよい。もし回転が、90度又は180度のように大きな増分で起これば、プロセスは毎回実行されてよい。

【0033】次に、前のグラフィックス状態が再び呼び出され（ステップ520）、可視オブジェクトのリストが戻され（ステップ522）、その後でプロセスは戻る。可視オブジェクトのこのリストは、シーン中のオブジェクトの入力リストを参照するインデックスの配列である。

【0034】再びステップ512を参照して、もしプロセスが継続の値を有しないならば、全てのオブジェクトが可視であると宣言され（ステップ514）、次にプロセスは前述したステップ518へ進む。

【0035】再びステップ508を参照して、もし新規シーン・フラグが`FALSE`であり、`get_new_occluders`が`FALSE`であれば、プロセスは前述したステップ512へ直接進む。

【0036】ここで図9を参照すると、遮断オブジェクトを選択するプロセスのフローチャートが、本発明の好ましい実施形態に従って描かれる。図9は、図8のステップ510を更に詳細に記述したものである。プロセスは、オブジェクトの可視領域、深度、及び複雑度値の加重合計に基づいて、オブジェクトをソートすることによって開始する（ステップ600）。これらの3つの属性の各々へ与えられたウェートは、実現方式に依存して変化してよい。描かれた例では、3つの属性の全てが、等しく加重される。オブジェクトの可視領域は、理想的には、その境界ボックスをビューポートへ投影することによってカバーされるピクセルの数の尺度である。オブジェクトの深度は、視点からオブジェクトの境界ボックス上の最も近い点までの`z`方向距離（正の`z`方向）の尺度である。複雑度は、オブジェクトを作り上げているプリミティブの数によって決定されてよく、更に、オブジェクトのテクスチャリングを考慮してよい。オブジェクトはソートされるから、各オブジェクトの可視領域値は、他の全てのオブジェクトの可視領域値にに関して正確であればよい。例えば、もし一組のオブジェクトの実際の可視領域が、10から100までの範囲を有するならば、実際の値を10で割った後では、同じ値が1から10までの範囲を有するであろう。可視領域値の双方の組の上でソートした結果は、オブジェクトの同じ順序を発生するであろう。

【0037】各オブジェクトの可視領域値は、他のオブジェクトの可視領域値に関してのみ正確であればよいかから、パフォーマンスを高めるために、可視領域を近似することができる。近似は、図8のステップ502で実行されるデータ構造の初期化の間に、各オブジェクトの境界ボックスにおける6つの面の3つの面領域を計算及びストアすることによって実行されてよい。3つの面の各々は、`x`、`y`、又は`z`座標軸のいずれかに垂直である。これらの領域値からベクタを形成することができる。次に、所与のビューにおけるオブジェクトの可視領域を、このベクタとアイ・ベクタとの積として近似することができる。その場合、各々の成分の積の絶対値が取られて合計される。

【0038】その後で、トータルのシーン複雑度の3.5%という経験的に決定された値より小さなトータル複雑度を有する最初の「X」個のオブジェクトが、ソートされたリストから選択される（ステップ602）。

「X」は、図2のプロセッサ202のパフォーマンス、図3のグラフィックス処理システム300、及びレンダリングされているオブジェクトのタイプに基づいて選択されなければならない。この例では、「X」は後述するように255を超えることはできない。

【0039】次に、「X」個のオブジェクトの各々は、独特の色を割り当てられ、その境界ボックスは、その色で深度データと共にバック・バッファ内の小さなビューポートへレンダリングされるが、ライティングはなされない（ステップ604）。このビューポートは「遮断オブジェクト選択ビューポート」である。次に、カラー・バッファがこのビューポートから読み出され、どの色がバッファ内に現れるかに従って、可視オブジェクトが決定される（ステップ606）。広い範囲のグラフィックス処理システムと互換性を持たせるため、本発明は、グラフィックス・システムによって少なくとも256色（8ビット・カラー）を表せるものと仮定する。背景を表すために1つの色を使用すれば（そこでは、オブジェクトはレンダリングされない）、区別することのできるオブジェクトは、最大で255個となる。各ピクセルがバッファから読み出されるとき、その色は、オブジェクトを表すか空の空間を表すかを決定する。もしそれがオブジェクトを表すならば、そのオブジェクトは遮断オブジェクトとしてフラグされる。

【0040】遮断オブジェクトとしてフラグされた全てのオブジェクトの複雑度値の合計を表す変数（`occluder_complexity`）、及び遮断オブジェクトとしてフラグされない全てのオブジェクトの複雑度値の合計を表す変数（`occludee_complexity`）の2つが管理される。もし`occluder_complexity`とウェート・ファクタの積が、`occluder_complexity`の値を超過すれば、他の変数`cull_worthy`が「TRUE」へセットされる（ステップ608）。このウ

エート・ファクタは、経験的に定義された他の値である。もし`cull_worthy`が、ピクセル読み出しプロセス中の成る時点で`FALSE`へセットされれば、本発明を継続するよりも、全てのオブジェクトを描画する方がより効率的であると決定される。その後で、プロセスは戻るか終了する。

【0041】図10を参照すると、遮断テストのプロセスのフローチャートが、本発明の好ましい実施形態に従って描かれる。図10は、図8のステップ516を更に詳細に記述したものである。

【0042】プロセスは、オブジェクトの入力配列中で最初のオブジェクトを検索することによって開始する（ステップ700）。検索されたオブジェクトは、遮断オブジェクト、及び遮断オブジェクトとして識別されなかつた他のオブジェクトの双方を含んでよい。従って、このオブジェクトが遮断オブジェクトとしてフラグされたかどうかの決定がなされる（ステップ702）。もしオブジェクトが遮断オブジェクトでなければ、新しい境界ボックスが現在のビューに対して計算される（ステップ704）。オリジナルの境界ボックスは、大域座標系（global coordinate system）に対するものである。しかし、ビューは、大域座標系と軸的に整列していないかも知れない。その結果、ビューと軸的に整列する新しい境界ボックスが計算され、その前壁（視点に最も近い面）がビュー平面と平行になるようにされる。その後で、プロセスは、新しい境界ボックスの前壁の深度及び投影を計算して、図8のステップ515でレンダリングされた遮断オブジェクトの深度を含むバック・バッファ内の小さなビューポート（遮断検出ビューポート）の中へ入れる（ステップ706）。この投影は、壁の投影された境界ボックスである。

【0043】その後で、前壁の深度が、壁の投影された境界ボックス内に存在するピクセルの深度と比較される（ステップ708）。これらの深度は、遮断オブジェクトの深度であり、図8のステップ515で得られたものである。もし前壁の深度が、壁の投影された境界ボックス内の全てのピクセルの深度値よりも大きければ、オブジェクトは遮断されるべきものと決定される（ステップ710）。もし全体のオブジェクトが遮断されるのであれば、オブジェクトは遮断されるものとしてラベルを付けられるだけである。そうでなければ、オブジェクトは可視である（ステップ712）。いずれの場合でも、次に、オブジェクトのチェックが終了したかどうかの決定がなされる（ステップ714）。もしオブジェクトのチェックが終了していなければ、プロセスは、次のオブジェクトをゲットし（ステップ716）、次にプロセスは前述したステップ702へ戻る。そうでなければ、プロセスは戻る。

【0044】再びステップ702を参照して、もしオブジェクトが遮断オブジェクトであれば、プロセスは前述

したステップ714へ直接進行する。

【0045】ここで図11～図14を参照すると、可視オブジェクトを識別するために使用される疑似コードの図が、本発明の好ましい実施形態に従って描かれる。図11～図14に示される疑似コードは、シーン内のどのオブジェクトが所与のビューから可視であるかを決定するために使用される。更に、このプロセスは、オブジェクトが可視オブジェクトによってブロックされるために描画される必要がないオブジェクトを識別する。プロセスが最も役立つのは、プロセスの実行時間プラス可視オブジェクトのレンダリング時間が、通常はシーンのために描画される全てのオブジェクトのレンダリング時間よりも小さいときである。

【0046】疑似コード800において、図11のセクション802は、入力及び出力データの識別を含む。オブジェクトの表示リストの配列、オブジェクトの境界ボックスの配列、オブジェクトの複雑度の配列、配列の長さ（入力オブジェクトのトータル数に等しい）、新しいシーンのためのフラグ（プロセスが実行された前の時間以後に、表示リストが変更されたときのみ`TRUE`へセットされる）、及びビューポートが、疑似コード800によって使用される入力データである。出力は、表示リストの入力配列を参照する可視オブジェクト・インデックスの配列、及びこの配列の長さ（可視オブジェクトの数に等しい）である。セクション802では、新しいシーンが描画されているかどうかの決定がなされる。もし新しいシーンが描画されていれば、データ構造がオブジェクトの各々のために作成される。これらのデータ構造は、ポリゴンの数、表示リスト・ポインタ、可視フラグ、最小及び最大頂点を含む境界ボックス、境界ボックス領域データ、及びデータ構造のオリジナルの表示リスト配列インデックスである。

【0047】セクション804は、遮断オブジェクトの選択及び遮断テストのような各種のサブルーチンを呼び出す命令を含む。これらは図13のセクション806及び図14の808により詳細に示される。図12のセクション804のコードは、現在のグラフィックス状態を保存し、遮断オブジェクトを選択し、遮断をテストして可視オブジェクトを識別し、グラフィックス状態をリストアするために使用される。図13のセクション806では、遮断オブジェクトを識別するために、また、選択された遮断オブジェクトの複雑度及び残りのオブジェクトの複雑度に基づいて、プロセスを継続すべきかどうかを決定するために、疑似コードが使用される。図14のセクション808にある疑似コードは、遮断をテストして、どのオブジェクトが遮断されるかを識別するために使用される。部分的に可視のオブジェクトは、本発明の好ましい実施形態に従って可視オブジェクトとして識別される。全面的に遮断されるオブジェクトのみが、遮断されるオブジェクトとして識別される。セクション80

8内のコードは、セクション810に示されるコードを呼び出す深度テストの呼び出しを含む。このコードは、深度バッファ内で投影された境界ボックス内の各ピクセルの深度を識別するために使用される。

【0048】ここで図15及び図16を参照すると、遮断されるオブジェクトを識別するために、本発明の遮断選別プロセスを実行すべきかどうかを決定するために使用される疑似コードの例が、本発明の好ましい実施形態に従って描かれる。疑似コード900は、セクション902の中に、オブジェクトが交差するか、又は完全にピュー・フラスタムの中にあるかどうかを決定する命令を含む。その後で、本発明の遮断選別プロセスを使用すべきかどうかの決定が、セクション904の命令によってなされる。基本的には、これらの命令は、セクション906で、遮断プロセスを実行するために使用される時間量を計時又は測定する。可視オブジェクトをレンダリングすることのできるレートが、セクション910のコードによって識別されたように、可視オブジェクトのレンダリング時間がセクション908で識別される。遮断されるオブジェクトをレンダリングしないことによって節約される時間量が、セクション912で推定される。プロセスを次回に使用すべきかどうかの決定は、セクション914の命令によってなされる。

【0049】本発明のメカニズムを使用する良好な候補シーンを識別するために、適格性基準が使用される。プロセスは、遮断されるものと宣言された全てのオブジェクトのレンダリング時間よりも早く実行されなければならない。悪い候補シーンについては、プロセスは終了され、通常のレンダリングが実行される。

【0050】本発明のメカニズムの良好な候補は、そのトータル複雑度の或るパーセンテージが不明瞭なシーンである。そのパーセンテージは、グラフィック・ハードウェア、プロセッサ速度、グラフィックス・ポートでのOpenGL実現方式、観察されているオブジェクトのタイプ、及びタイミング・テストのタイプのような各種のファクタに基づく経験的な値である。

【0051】注意すべき重要なこととして、本発明は、完全に機能するデータ処理システムの関連で説明されたが、当業者は、本発明のプロセスが、命令のコンピュータ読み取り可能媒体の形式及び多様な形式で発布可能であること、また本発明は発布を実行するため実際に使用される信号搬送媒体の特定のタイプに関係なく等しく応用されることを了解するであろう。コンピュータ読み取り可能媒体の例は、フロッピー・ディスク、ハード・ディスク・ドライブ、RAM、及びCD-ROMのような記録可能形媒体、及び、デジタル及びアナログ通信リンクのような伝送形媒体を含む。

【0052】本発明の記述は、例示及び説明の目的で提示されたが、開示された形式の発明に全てが尽くされること、又は、限定されること、を意図するものではない。

い。多くの変更及びバリエーションが当業者に明らかであろう。描かれた例はオブジェクトの処理に向けられているが、本発明のプロセスは、ポリゴン又はブリミティブのような他の品目に応用されてよい。実施形態の選択と説明は、本発明の原理及び実際の応用を最も明らかにし、且つ当技術分野に通常の知識を有する他の人々が、想定される特定の使用に適した各種の変更を有する各種の実施形態のために本発明を理解できるようになされた。

【0053】まとめとして、本発明の構成に関して以下の事項を開示する。

(1) 視点のために描画を表示するデータ処理システムにおける方法であって、ここで描画は一組のオブジェクトを含み、該方法は、オブジェクトの組のために複数の境界ボックス及び複雑度データを受け取り、ここで前記境界ボックス及び前記複雑度データはオブジェクトの組内の各オブジェクトに関連づけられ、複数の前記境界ボックスを使用して前記オブジェクトの組内の遮断オブジェクトを選択し、視点のために可視のオブジェクトを前記オブジェクトの組から識別するコンピュータ実現ステップを含む方法。

(2) 前記オブジェクトが識別されたオブジェクトであり、前記識別されたオブジェクトを表示するステップを更に含む、上記(1)に記載の方法。

(3) ピクセル深度データが各前記遮断オブジェクトに関連づけられ、且つ可視であるオブジェクトの組からオブジェクトを識別するステップが、前記オブジェクトの組内のオブジェクトの前記境界ボックスの深度が、前記遮断オブジェクトのピクセルの深度よりも視点に近いかどうかを決定することを含む、上記(1)に記載の方法。

(4) 前記ピクセル深度データが、前記ピクセル深度データを発生するための低解像度で深度情報を用いて前記遮断オブジェクトを十分詳細にレンダリングすることによって発生する、上記(3)に記載の方法。

(5) 前記遮断オブジェクトを選択するステップが、前記オブジェクトの組を複雑度によってソートし、前記オブジェクトの組から、描画の複雑度の設定パーセンテージよりも小さなトータル複雑度を有する或る数のオブジェクトを選択することを含む、上記(1)に記載の方法。

(6) 前記オブジェクトの数が255よりも少ない、上記(5)に記載の方法。

(7) 前記設定パーセンテージが3.5パーセントである、上記(6)に記載の方法。

(8) 前記推定するステップが、前記オブジェクトの組の中で、全体的な最大可視領域、最も近い深度値、及び最低複雑度値を有するオブジェクトを識別することを含む、上記(1)に記載の方法。

(9) 前記複雑度データが、オブジェクトをレンダリン

グするのに必要な時間を含む、上記（1）に記載の方法。

（10）前記遮断オブジェクトのトータル複雑度が残りの部分の複雑度よりも大きいかどうかを決定することを更に含む、上記（1）に記載の方法。

（11）前記他のオブジェクトが可視であるかどうかを決定するステップが、前記遮断オブジェクトの複雑度が残りの部分の複雑度よりも小さいことの決定に応答する、上記（10）に記載の方法。

（12）選択、識別、及び表示ステップを実行するために使用される第1の時間量を決定し、選択及び識別ステップを実行することなくオブジェクトの組を表示するために使用される第2の時間量を決定し、第1の時間量が第2の時間量よりも大きいことに応答して、オブジェクトの組を表示するとき選択及び識別ステップの実行を防止することを更に含む、上記（2）に記載の方法。

（13）視点のためにシーンを処理するデータ処理システムにおける方法であって、ここでシーンは複数のオブジェクトを含み、該方法は、シーンのためのデータを受け取り、シーンのためのデータを使用して前記複数のオブジェクトから一組の可視オブジェクトを選択し、前記複数のオブジェクトから選択されないオブジェクトを、前記可視オブジェクトの組と比較することによって、視点のために前記可視のオブジェクトを識別するデータ処理システムのステップを含む方法。

（14）データが、前記複数のオブジェクトの境界ボックス、及び前記複数のオブジェクトの複雑度データを含む、上記（13）に記載の方法。

（15）ピクセル深度データが各前記可視オブジェクトに関連づけられ、オブジェクトの組からの他のオブジェクトが可視であるかどうかを識別するステップが、前記境界ボックスの深度が前記遮断オブジェクトのピクセルの深度よりも視点に近いかどうかを決定することを含む、上記（14）に記載の方法。

（16）視点のために描画を表示するデータ処理システムであって、ここで描画は一組のオブジェクトを含み、該データ処理システムは、オブジェクトの組のために複数の境界ボックス及び複雑度データを受け取る受け取り手段と、ここで前記境界ボックス及び前記複雑度データはオブジェクトの組内の各オブジェクトに関連づけられ、複数の前記境界ボックスを使用して、前記オブジェクトの組内の遮断オブジェクトを選択する選択手段と、前記オブジェクトの組から、視点のために可視のオブジェクトを識別する識別手段とを含むデータ処理システム。

（17）前記オブジェクトが識別されたオブジェクトであり、前記識別されたオブジェクトを表示する表示手段を更に含む、上記（16）に記載のデータ処理システム。

（18）ピクセル深度データが各前記遮断オブジェクト

に関連づけられ、且つ前記可視のオブジェクトをオブジェクトの組から識別するステップが、前記オブジェクトの組内のオブジェクトの境界ボックスの深度が、前記遮断オブジェクトのピクセルの深度よりも視点に近いかどうかを決定する決定手段を含む、上記（16）に記載のデータ処理システム。

（19）前記ピクセル深度データが、ピクセル深度データを発生するための低解像度で深度情報を用いて前記遮断オブジェクトを十分詳細にレンダリングすることによって発生する、上記（18）に記載のデータ処理システム。

（20）前記遮断オブジェクトを選択する選択手段が、前記オブジェクトの組を複雑度によってソートするソート手段と、前記オブジェクトの組から、描画の複雑度の設定パーセンテージよりも小さいトータル複雑度を有する或る数のオブジェクトを選択する選択手段とを含む、上記（16）に記載のデータ処理システム。

（21）前記オブジェクトの数が255より少ない、上記（20）に記載のデータ処理システム。

（22）前記設定パーセンテージが3.5パーセントである、上記（21）に記載のデータ処理システム。

（23）前記推定手段が、前記オブジェクトの組の中で、全体的な最大可視領域、最も近い深度値、及び最低複雑度値を有するオブジェクトを識別する識別手段を含む、上記（16）に記載のデータ処理システム。

（24）前記複雑度データが、オブジェクトをレンダリングするに必要な時間を含む、上記（16）に記載のデータ処理システム。

（25）前記遮断オブジェクトのトータル複雑度が残りの部分の複雑度よりも大きいかどうかを決定する決定手段を更に含む、上記（16）に記載のデータ処理システム。

（26）前記他のオブジェクトが可視であるかどうかの決定手段が、前記遮断オブジェクトの複雑度が残りの部分の複雑度よりも小さいことの決定に応答する、上記（25）に記載のデータ処理システム。

（27）選択、識別、及び表示ステップを実行するために使用される第1の時間量を決定する第1の決定手段と、選択、及び識別ステップを実行することなく前記オブジェクトの組を表示するために使用される第2の時間量を決定する第2の決定手段と、前記第1の時間量が前記第2の時間量よりも大きいことに応答して、前記オブジェクトの組を表示するとき選択及び識別手段の制限を防止する防止手段とを更に含む、上記（17）に記載のデータ処理システム。

（28）視点のためにシーンを処理するデータ処理システムであって、ここでシーンは複数のオブジェクトを含み、該データ処理システムは、シーンのためのデータを受け取る受け取り手段と、シーンのためのデータを使用して、複数のオブジェクトから一組の可視オブジェクト

を選択する選択手段と、前記複数のオブジェクトから選択されないオブジェクトを、前記可視オブジェクトの組と比較することによって、視点のために前記可視のオブジェクトを識別する識別手段とを含む。

(29) 前記データが、複数のオブジェクトのための境界ボックス、及び複数のオブジェクトのための複雑度データを含む、上記(28)に記載のデータ処理システム。

(30) ピクセル深度データが各可視オブジェクトに関連づけられ、前記オブジェクトの組からの他のオブジェクトが可視であるかどうかを識別するステップが、前記境界ボックスの深度が、遮断オブジェクトのピクセルの深度よりも視点に近いかどうかを決定する決定手段を含む、上記(29)に記載のデータ処理システム。

(31) 視点のために描画を表示するための、コンピュータ読み取り可能媒体におけるコンピュータ・プログラム製品であって、ここで描画は一組のオブジェクトを含み、該コンピュータ・プログラム製品は、オブジェクトの組のために複数の境界ボックス及び複雑度データを受け取る第1の命令と、ここで前記境界ボックス及び前記複雑度データは、オブジェクトの組の中の各オブジェクトに関連づけられ、複数の前記境界ボックスを使用して前記オブジェクトの組の中の遮断オブジェクトを選択する第2の命令と、前記オブジェクトの組から、視点のために可視のオブジェクトを識別する第3の命令とを含むコンピュータ・プログラム製品。

(32) 視点のためにシーンを処理するための、コンピュータ読み取り可能媒体におけるコンピュータ・プログラム製品であって、ここで前記シーンは複数のオブジェクトを含み、前記コンピュータ・プログラム製品は、シーンのためのデータを受け取る第1の命令と、シーンのためのデータを使用して、複数のオブジェクトから一組の可視オブジェクトを選択する第2の命令と、前記複数のオブジェクトから選択されないオブジェクトを、前記可視オブジェクトの組と比較することによって、視点のために可視のオブジェクトを識別する第3の命令とを含むコンピュータ・プログラム製品。

【図面の簡単な説明】

【図1】本発明が実現されてよいデータ処理システムを描いた図である。

【図2】本発明が実現されてよいデータ処理システムを示すブロック図である。

【図3】本発明の好ましい実施形態に従って描かれたグラフィックス処理システムのシステム・アーキテクチャを示すブロック図である。

【図4】本発明の好ましい実施形態に従って描かれたビュー・フラスタムにおけるオブジェクトの展望図である。

【図5】本発明の好ましい実施形態に従って描かれたビュー・フラスタムにおけるオブジェクトの展望図である。

る。

【図6】本発明の好ましい実施形態に従って描かれたビュー・フラスタムにおけるオブジェクトの展望図である。

【図7】本発明の好ましい実施形態に従って描かれたビュー・フラスタムにおけるオブジェクトの展望図である。

【図8】本発明の好ましい実施形態に従って描かれた所与のビューから見ることのできるオブジェクトの最小の組を識別するプロセスを示すフローチャートである。

【図9】本発明の好ましい実施形態に従って描かれた遮断オブジェクトを選択するプロセスを示すフローチャートである。

【図10】本発明の好ましい実施形態に従って描かれたオブジェクトの遮断をテストするプロセスを示すフローチャートである。

【図11】本発明の好ましい実施形態に従って描かれた可視オブジェクトを識別するために使用される疑似コードを示す図である。

【図12】本発明の好ましい実施形態に従って描かれた可視オブジェクトを識別するために使用される疑似コードを示す図である。

【図13】本発明の好ましい実施形態に従って描かれた可視オブジェクトを識別するために使用される疑似コードを示す図である。

【図14】本発明の好ましい実施形態に従って描かれた可視オブジェクトを識別するために使用される疑似コードを示す図である。

【図15】本発明の好ましい実施形態に従って描かれた被遮断オブジェクトを識別するために本発明の遮断選別プロセスを実行すべきか否かを決定するために使用される疑似コードを示す図である。

【図16】本発明の好ましい実施形態に従って描かれた被遮断オブジェクトを識別するために本発明の遮断選別プロセスを実行すべきか否かを決定するために使用される疑似コードを示す図である。

【符号の説明】

100 パーソナル・コンピュータ

102 ビデオ表示端末

104 キーボード

106 マウス

108 記憶装置

110 システム・ユニット

200 データ処理システム

202 プロセッサ

204 メイン・メモリ

206 PCIローカル・バス

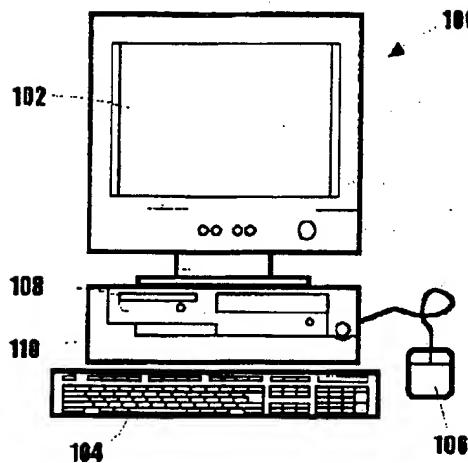
208 PCIブリッジ

210 ローカル・エリア・ネットワーク (LAN) アダプタ

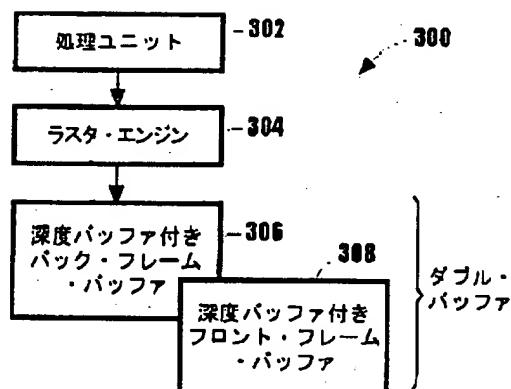
212 SCS I ホスト・バス・アダプタ
 214 拡張バス・インターフェース
 216 オーディオ・アダプタ
 218 グラフィックス・アダプタ
 219 オーディオ/ビデオ・アダプタ
 220 キーボード及びマウス・アダプタ
 222 モデム
 224 追加メモリ
 226 ハード・ディスク・ドライブ
 228 テープ・ドライブ
 230 CD-ROM ドライブ

300 グラフィックス処理システム
 302 処理ユニット
 304 ラスター・エンジン
 306 バック・フレーム・バッファ
 308 フロント・フレーム・バッファ
 400 フラスタム
 412 視点
 414 近い平面
 416 遠い平面
 800 疑似コード
 900 疑似コード

【図1】

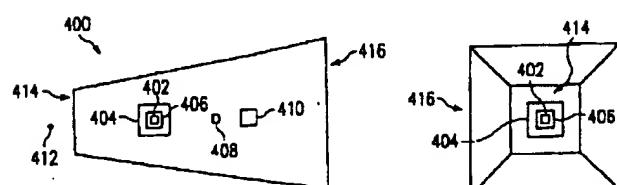
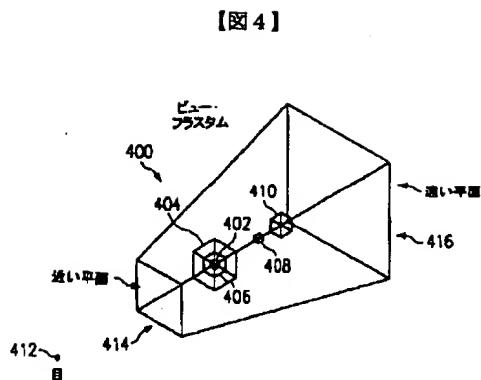


【図3】

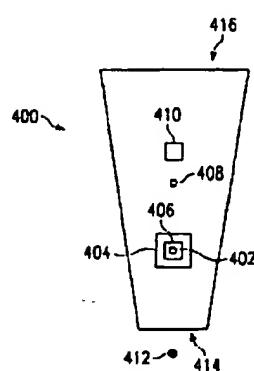


【図5】

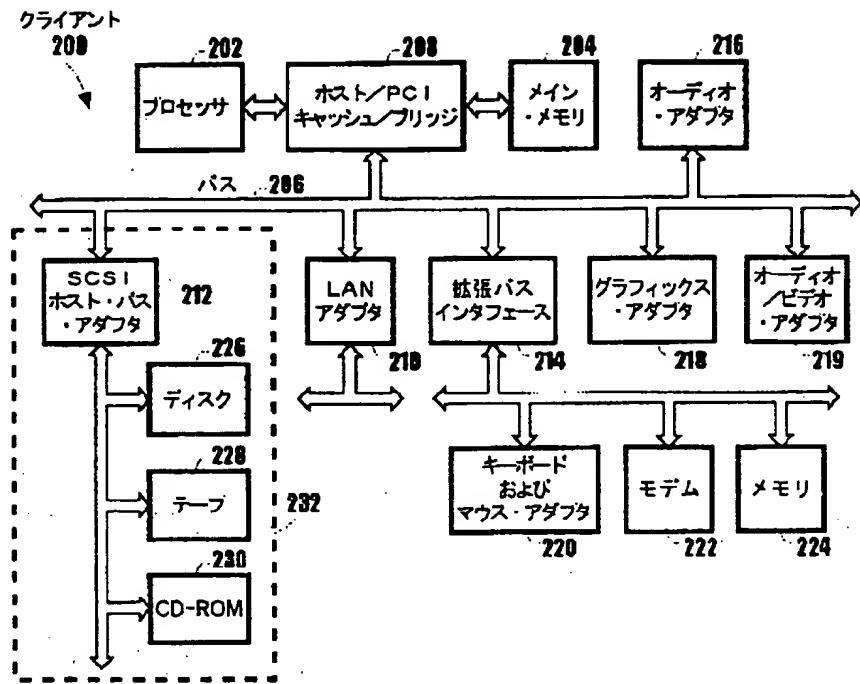
【図6】



【図7】



[图2]

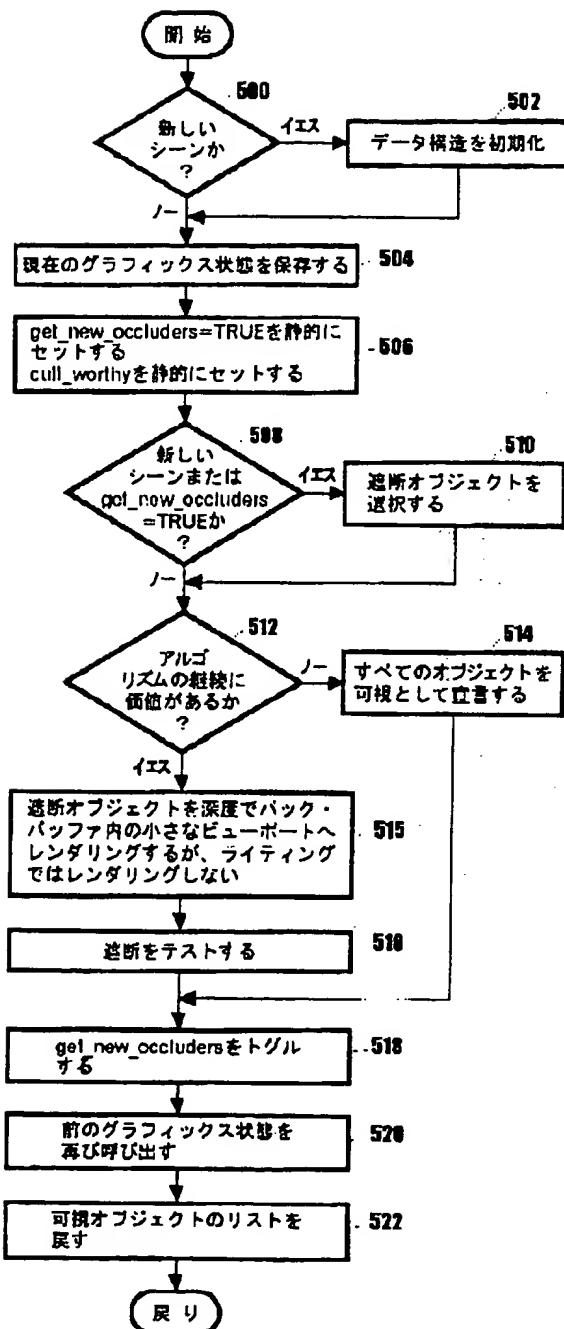


[图 15]

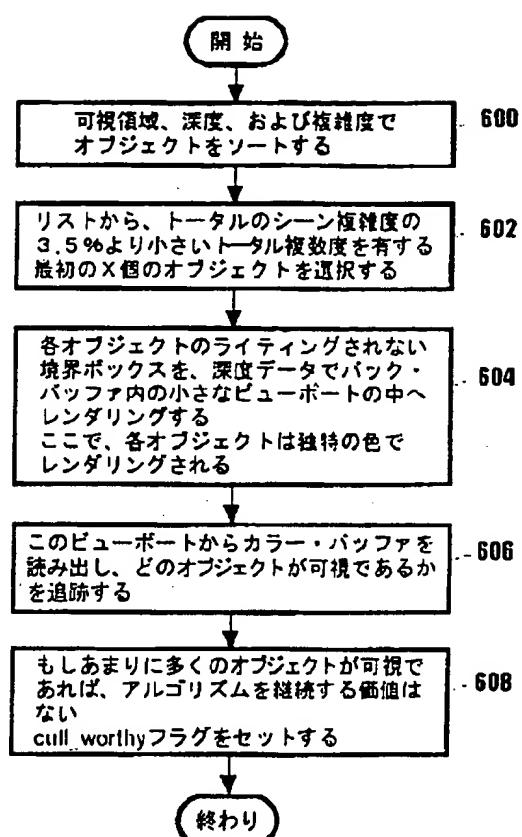
```
// 最初のステップは、どのオブジェクトが交差するか、または
// 完全にビュー・フラスタムの中にあるかを決定することである
for(i=0; i<nObject; i++) {
    if ( !objectOutsidesViewFrustum( object [i] ) ) {
        viewableObjects[nViewable] = object [i]
        viewableBoundingBoxes[nViewable] = objectBoundingBox[i]
        viewableComplexities[nViewable] = objectComplexity[i]
        viewableComplexity += objectComplexity[i]
        nViewable++
    }
}
```

図 16 ~

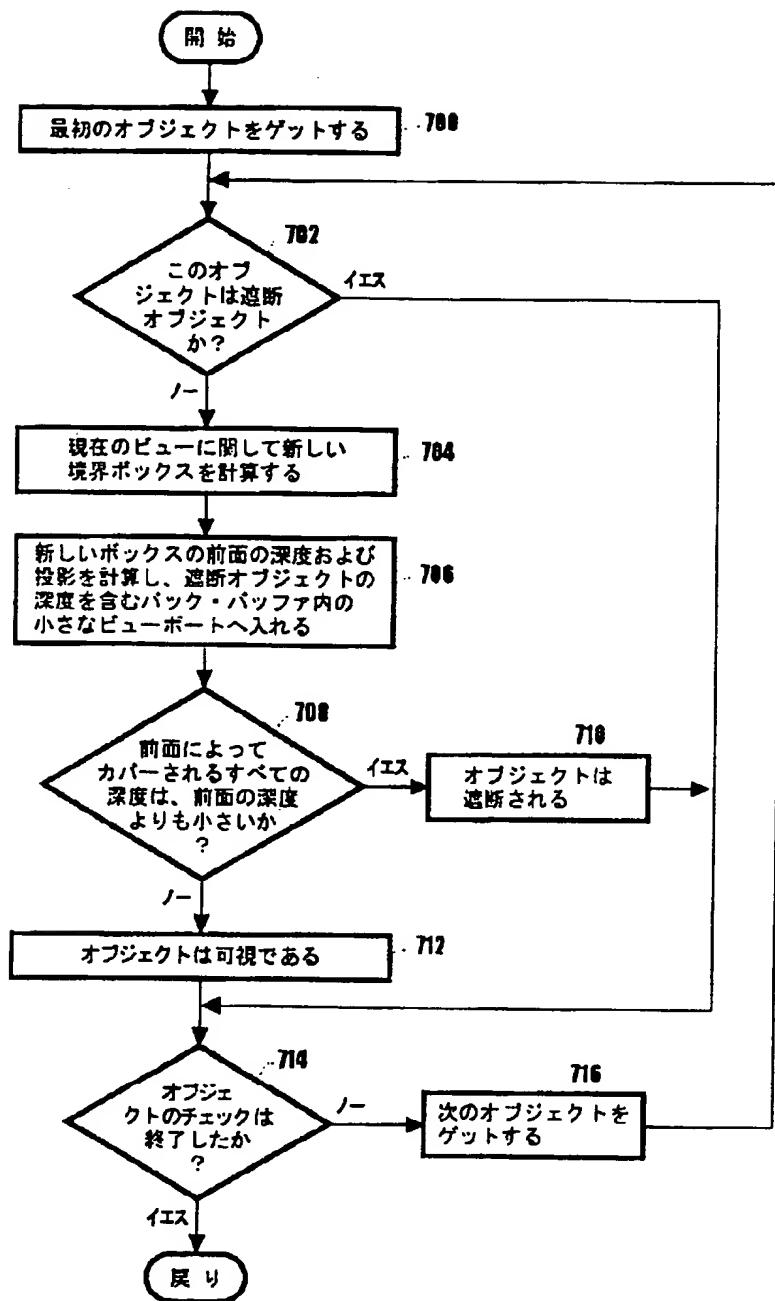
【図8】



【図9】



【図10】



【図11】

cull

データ

入力： オブジェクトの表示リストの配列
オブジェクトの境界ポックスの配列
オブジェクトの複雑度の配列
配列の長さ (=N)
もし表示リストが前の呼び出しから変化していれば、
フラグ (新規シーン・フラグ)
ビューポート

出力： 入力オブジェクト配列への、可視オブジェクトを表す
インデックスの配列
配列の長さ

メソッド

if (新しいシーン) {
 create N個の表示リストに対応するN個のオブジェクトを
 作成する;
 initialize 各オブジェクトを次のもので初期化する {
 表示リスト内のホリゴンの数(複雑度);
 表示リスト・ポインタ;
 可視フラグ;
 境界ポックスの最小および最大頂点;
 境界ポックスの領域データ;
 オリジナルの表示リスト配列インデックス数;
 }
}
call occlusion_cullを呼び出す — returns 可視オブジェクトの
リストを戻す;
return 戻る;

800

802

図12へ

【図12】

図11から

occlusion_cull
=====

データ

入力： オブジェクトの配列
配列の長さ
新規シーン・フラグ
ビューホート
遮断オブジェクトの選択におけるミニ・ビューポートの解像度
被遮断オブジェクトのテストにおけるミニ・ビューポートの解像度
注：一般的に、被遮断オブジェクトのミニ・ビューポートの
解像度は、遮断オブジェクトのビューポートの解像度
よりも大きい

出力： 可視オブジェクト・インデックスの配列
配列の長さ

メソッド

save 現在のグラフィックス状態を保存する;
disable ディザリングを抑制する;
concatenate ビューおよび投影マトリックスを連結する;
set フラグ get_new_occluders = FALSE をセットする;
if (新しいシーン OR get_new_occluders)
 call pic_occluders を呼び出す — returns 遮断オブジェクトの
 リストおよび cull_worthy フラグを戻す;
 get_new_occluders = !get_new_occluders;
 if cull_worthy {
 clear パック・バッファ内のミニ・ビューポートの深度を
 クリアする
 disable ライティングを不可能にする
 render 遮断オブジェクトをパック・バッファのミニ・ビューポート
 へレンダリングする
 enable ライティングを可能にする
 call test_occlusion を呼び出す … returns 可視オブジェクト
 のリストを戻す;
 }
 else
 draw すべてを描画する;
 restore 前のグラフィックス状態をリストアする;
 return 戻る;

808

804

【図13】

```

pick_occluders
=====
データ
-----
    入力： 遮断オブジェクトのミニ・ビューポートの解像度
          ビューポート
          オブジェクトの配列
          配列の長さ
          モデルビューのマトリックス
          モデルビューおよび投影マトリックスの連結

    出力： 遮断オブジェクトの配列
          配列の長さ
          cull_worthyフラグ (もしアルゴリズムを進める値があればTRUE、
          もしけてを描画するのが早ければFALSE)

メソッド
-----
    for 各オブジェクトについて {
        compute その境界ボックスの可視領域を、現在のビューから計算する;
        compute Z方向で、その境界ボックスのより近い側面の深度を計算する;
    }
    store シーン内のオブジェクトの全リストについて、可視領域の範囲、深度、
          および複雑度をストアする;
    for 各オブジェクトについて
        assign 可視領域、深度、および複雑度に基づいてウェートを割り当てる;
        sort オブジェクトをウェートでソートする;
        pick ソートされたリストから最初のX個のオブジェクトを、可能性のある
              遮断オブジェクトとして選択する。Xは経験的に決定される。
              可能性のある遮断オブジェクトは、シーンのトータル複雑度の3.5%よりも
              小さなトータル複雑度を有しなければならない。
              さらに、Xは255を超えることはできない;
        clear 遮断オブジェクトのみに、ビューポートのバック・バッファ内の色
              および深度をクリアする;
        for 可能性のある各遮断オブジェクトについて {
            set 現在の色を、オブジェクトのインデックスに対応する独特の値へ設定する;
            render オブジェクトのライティングされない境界ボックスを、深度で
                  遮断オブジェクトのミニ・ビューポートのバック・バッファへ
                  レンダリングする;
        }
        read カラー・バッファを遮断オブジェクトのミニ・ビューポートから読み出す;
        define オブジェクトを認識するために必要なピクセルの最小数を、
              mini_num_pixels = 0.001*遮断オブジェクトのミニ・ビューポート中の
              ピクセル数として定義する;
        for 遮断オブジェクトのミニ・ビューポート内の各ピクセルについて {
            when オブジェクトが発見されたとき {
                // このオブジェクトが遮断オブジェクトとして既にフラグされていなければ {
                    flag オブジェクトを遮断オブジェクトとしてフラグする;
                    keep すべての遮断オブジェクトのトータル複雑度を追跡する;
                    check 横向の値があるかどうかをチェックする
                        (cull_worthy = occluder_complexity * weight_factor
                         > occcluder_complexity)
                }
            }
        }
    }
    // 遮断オブジェクトが発見されなければ(オブジェクトがあまりに小さく縮尺されて
    // いるかもしれない)、ソートされたリスト中の最初のオブジェクトを唯一の遮断
    // オブジェクトとして使用する
    return 戻る;

```

898

896

【図14】

```
test_occlusion
=====
データ
_____
入力： 被遮断オブジェクトのテストにおけるミニ・ビューポートの解像度
ビューポート
モデルビューおよび投影マトリックスの連結
オブジェクトの配列
配列の長さ
被遮断オブジェクトのミニ・ビューポート

出力： 可視オブジェクト・インデックスの配列
配列の長さ

メソッド
_____
create 被遮断オブジェクトのミニ・ビューポートから読み出された
深度データを保持するために、2Dバッファを作成する;
for 可能性のある各被遮断オブジェクトについて {
    project オブジェクトの境界ボックスをメイン・ビューポートへ
    投影する(深度情報を用いて);
    get スクリーン情報をゲットする;
    scale 座標を、被遮断オブジェクトのミニ・ビューポートへ
    合わせるために縮尺する;
    if これが1回目のループ通過であれば
        read 深度データをバッファへ読み出す;
        call depth_testを呼び出す — returns TRUE(遮断),
        またはFALSE(可視)を戻す;
    if 遮断されなければ
        flag オブジェクトを可視としてフラグする;
}
return 戻る;

depth_test
=====
データ
_____
入力： 被遮断オブジェクトのミニ・ビューポートからの深度バッファ
ビューパlaneに平行なZ平面、およびそれぞれ垂直および
水平なXおよびY平面を有する3Dボックスの大きさ

出力： もしオブジェクトが遮断されるのであればTRUE、
または、もしオブジェクトが可視であればFALSE

メソッド
_____
for 被遮断オブジェクトのミニ・ビューポートからの深度バッファの
各ピクセルについて
if バッファ深度が、Z方向における境界ボックスの
より近い側面の深度よりも大きければ
    return FALSEを戻す;
return TRUEを戻す;
```

800

808

810

【図16】

図15から

```

// アルゴリズムを試みるべきかどうかを決定するために、前の情報を使用する。
if(!okayToOcclude) {
    // 遮断アルゴリズムを計時する
    T_ocb = Timer( occlusionCullBegin )
    doOcclusion( nViewable, viewableObjects, viewableBoundingBoxes,
        viewableComplexities, nVisible, visibleObjectIndices,
        T_oca = Timer( occlusionCullEnd )
    T_oc = T_oca - T_ocb
    // 可視オブジェクトのレンダリングを計時する
    T_rvb = Timer( renderVisibleBegin )
    908    visibleComplexity = renderVisible( nVisible, visibleObjectIndices,
        viewableObjects, viewableComplexities )
    T_rve = Timer( renderVisibleEnd )
    T_rv = T_rve - T_rvb
    910    // 可視オブジェクトがレンダリングされることのできるレートを計算する
    R_v = visibleComplexity / T_rv
    // 被遮断オブジェクトをレンダリングしないことによって節約される
    // 時間量を推定する
    912    occludedComplexity = viewableComplexity - visibleComplexity
    R_o = occludedComplexity / R_v
    904

    // アルゴリズムを次回に使用すべきかどうかを決定する
    if ( R_o > T_oc )
        okayToOcclude = True
    else {
        // 遮断アルゴリズムはあまりに長い時間をとった。
        // アルゴリズムを目的的に再び実行しないのではなく、
        // 決定を成り行きに任せる
        okayToOcclude = takechance
    }
    else {
        // 交差するか、または完全にビュー・フラスタムの内部にある
        // すべてのオブジェクトをレンダリングする
        renderVisible( nObject, object )
        // 次のフレームが遮断選別を試みるべきかどうかを
        // 計るため実行してみる
        okayToOcclude = takechance
    }
}

```

フロントページの続き

(72)発明者 ブラッドフォード・リー・コブ
 アメリカ合衆国78613 テキサス州、シ
 イダー・パーク、レッド・ランチ・サー
 クル 1231

(72)発明者 リー・マイケル・グロス
 アメリカ合衆国78727 テキサス州、オ
 ースティン、アレメア・トレース・サー
 クル 12320 #1009

THIS PAGE BLANK (USPTO)

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK (USPTO)